(12) UK Patent Application (19) GB (11) 2 387 710 (13) A

(43) Date of A Publication 22.10.2003

(21) Application No 0208795.5

(22) Date of Filing 17.04.2002

(71) Applicant(s)
Heriot-Watt University
(Incorporated in the United Kingdom)
Riccarton, EDINBURGH, EH14 4AS,
United Kingdom

(72) Inventor(s)
Miguel Arredondo
Douglas Marsland
David Michael Lane
Adriana Olmos
Alastair Cormack
Arnaud Schmidt
Katia Lebart

(74) Agent and/or Address for Service
Cruikshank & Fairweather
19 Royal Exchange Square, GLASGOW,
G1 3AE, United Kingdom

(51) INT CL^7
G11B 27/028

(52) UK CL (Edition V)
G5R RB81

(56) Documents Cited
GB 2380599 A
EP 0780844 A2
US 5956453 A
US 5177513 A
EP 0915469 A2
EP 0675495 A2
US 5339166 A

(58) Field of Search
UK CL (Edition V) **G5R**
INT CL^7 **G11B**
Other: **Online: EPODOC; JAPIO; WPI. Selected publications.**

(54) Abstract Title
Digital video editing

(57) A system for editing digital video footage using a computer processor. The system is configured to cut (102) the digital video footage (100) into a plurality of clips and sort (104) those clips into groups. A group identifier associated with each of the groups is presented on a screen or display (107). The groups can be selected by a user so that the clips contained therein can be used. The cutting can be done automatically using a cutting algorithm, eg based on a histogram, or by intensity edge determination. Clips may be classified according to predetermined criteria eg camera motion, optical flow, image quality, scene changes, photometric characteristics level of blurring and contrast detection.

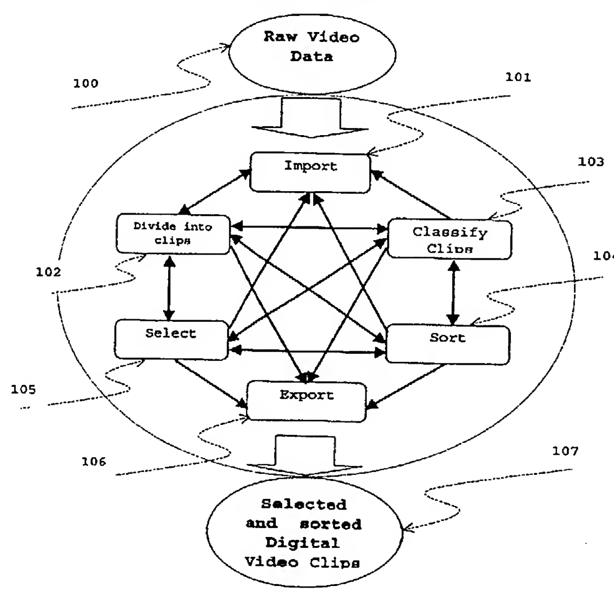

Figure 2

This print takes account of replacement documents submitted after the date of filing to enable the application to comply with the formal requirements of the Patents Rules 1995

GB 2 387 710 A

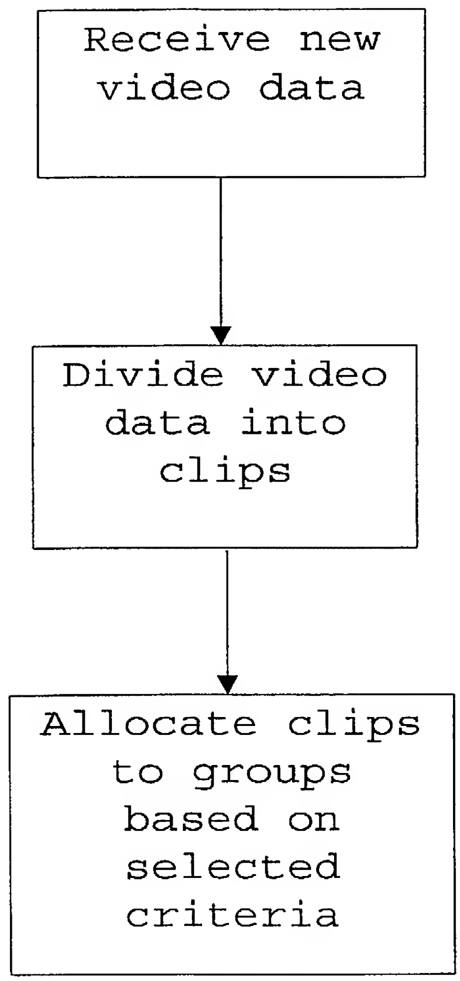
Receive new video data
Divide video data into clips
Allocate clips to groups based on selected criteria

Figure 1

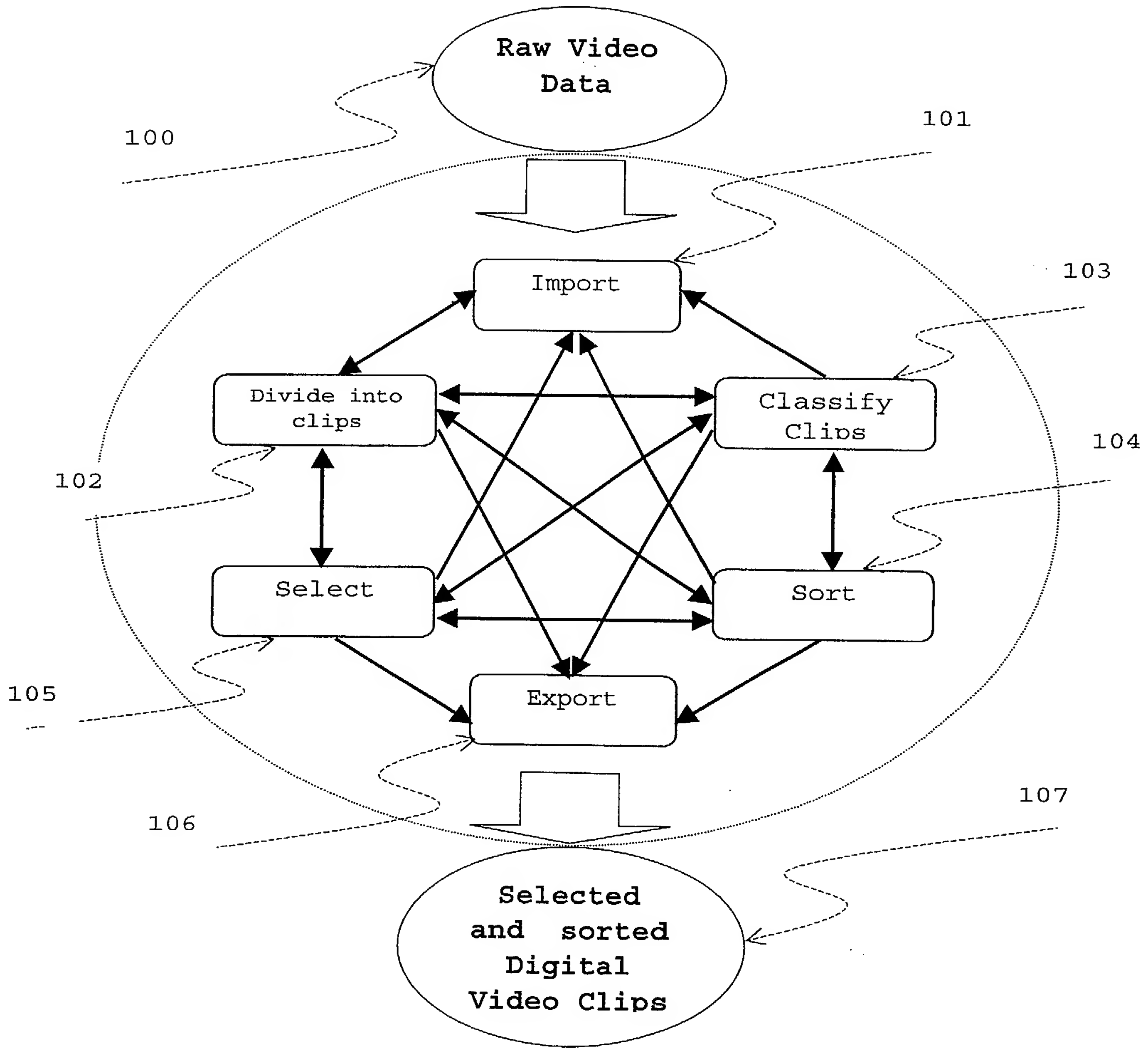
Raw Video Data
100
101
Import
103
Divide into clips
Classify Clips
104
102
Select
Sort
105
Export
106
107
Selected and sorted Digital Video Clips

Figure 2

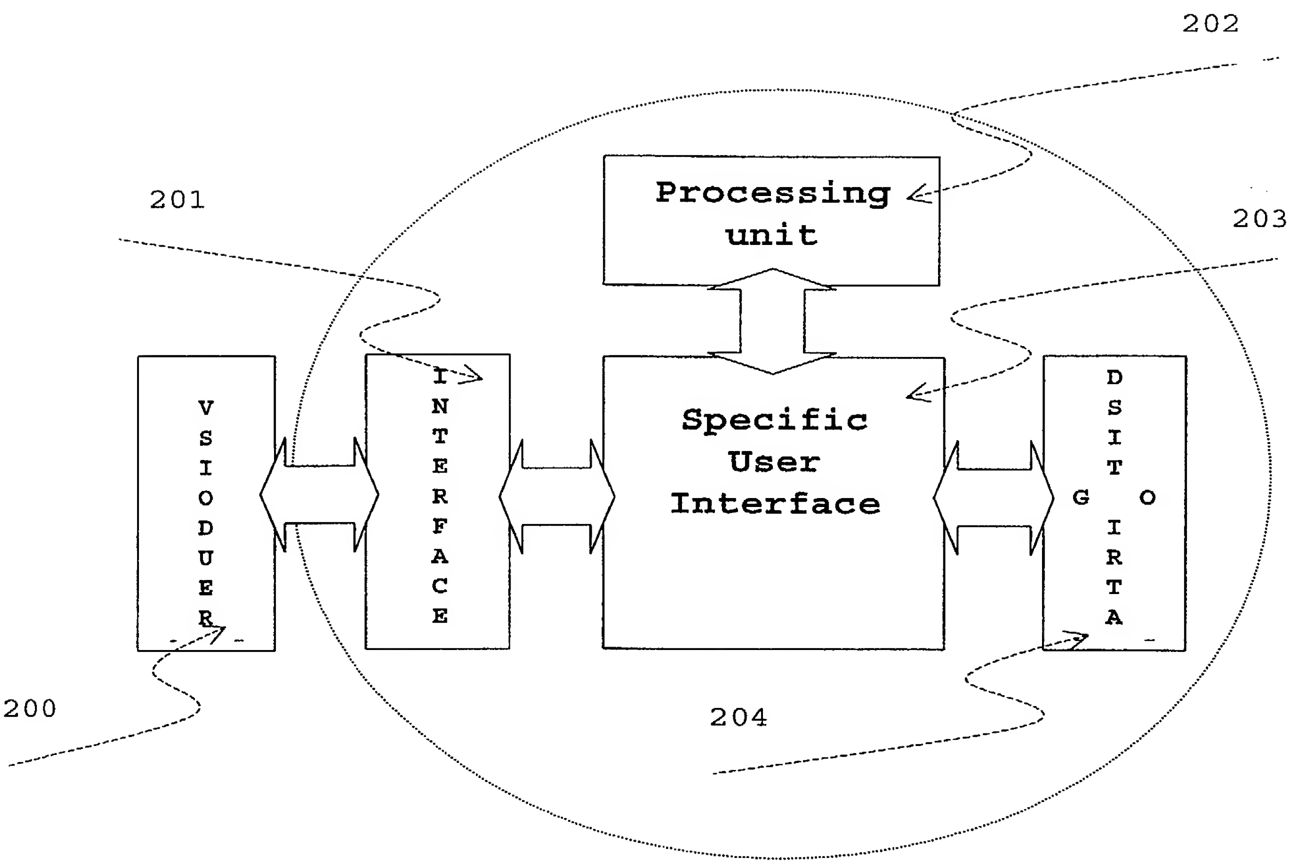
202
201
203
Processing unit
Specific User Interface
V S I O D U E R
I N T E R F A C E
D S I T G O I R T A
200
204

Figure 3

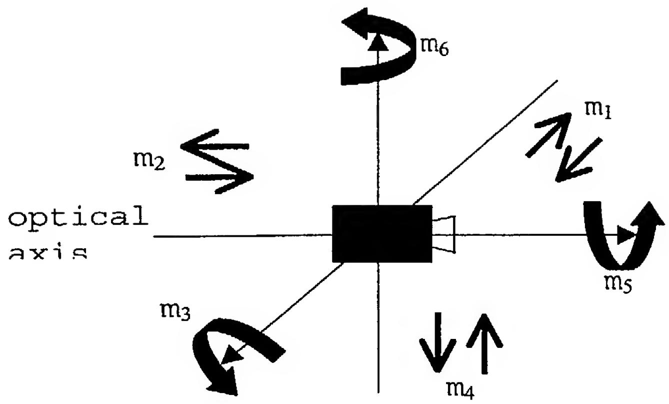

Figure 4

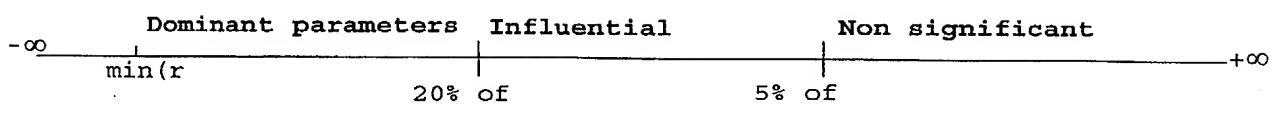
Dominant parameters
Influential
Non significant
-∞
+∞
min(r
20% of
5% of

Figure 5

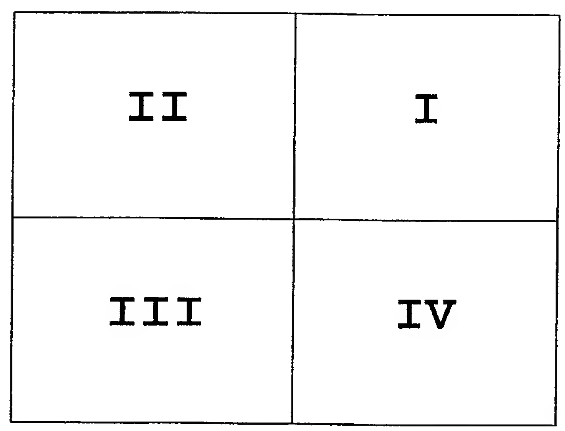

Figure 6

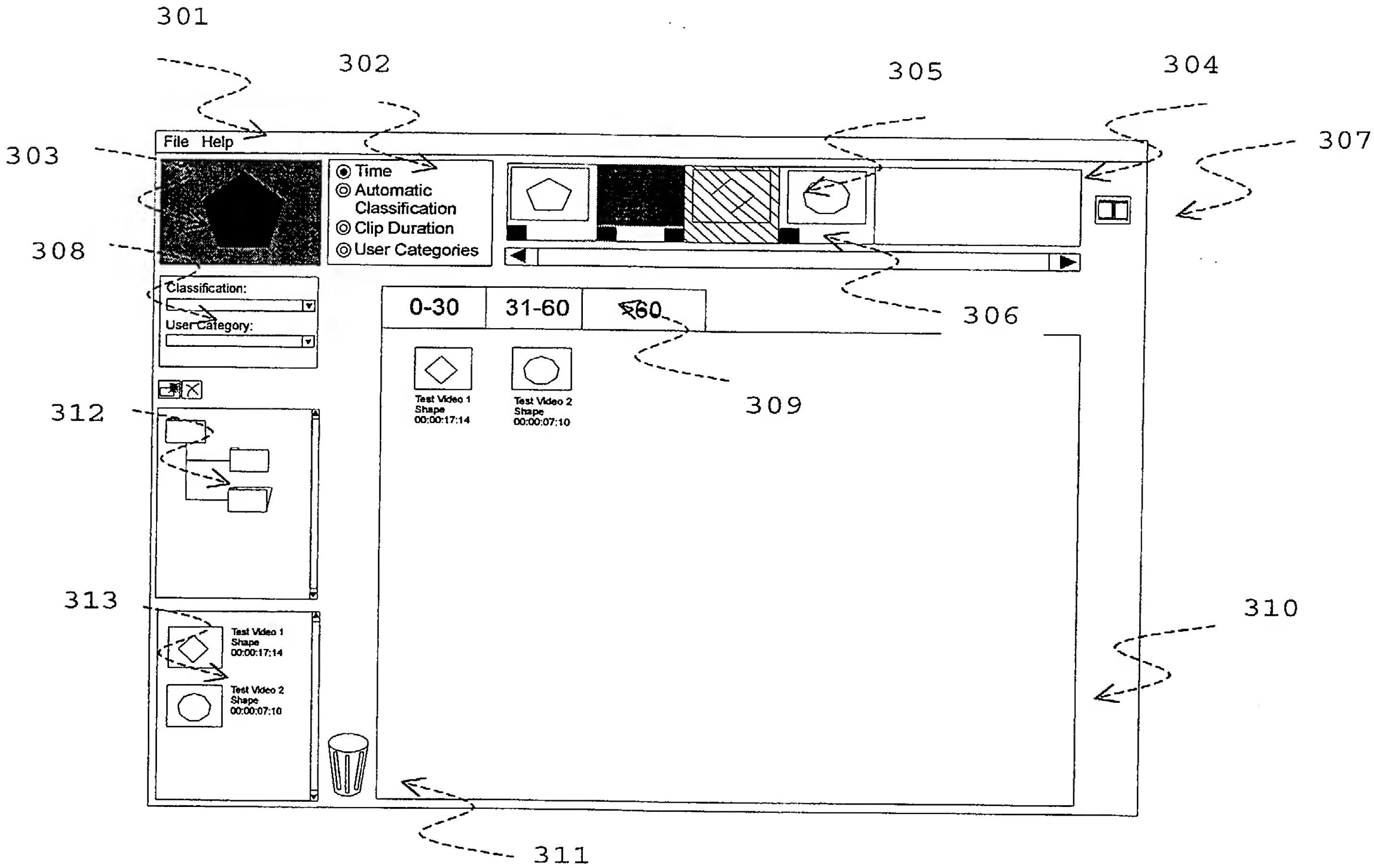
301
302
305
304
307
303
308
306
309
312
313
310
311
File Help
Time
Automatic Classification
Clip Duration
User Categories
Classification:
User Category:
0-30
31-60
>60
Test Video 1
Shape
00:00:17:14
Test Video 2
Shape
00:00:07:10

Figure 7

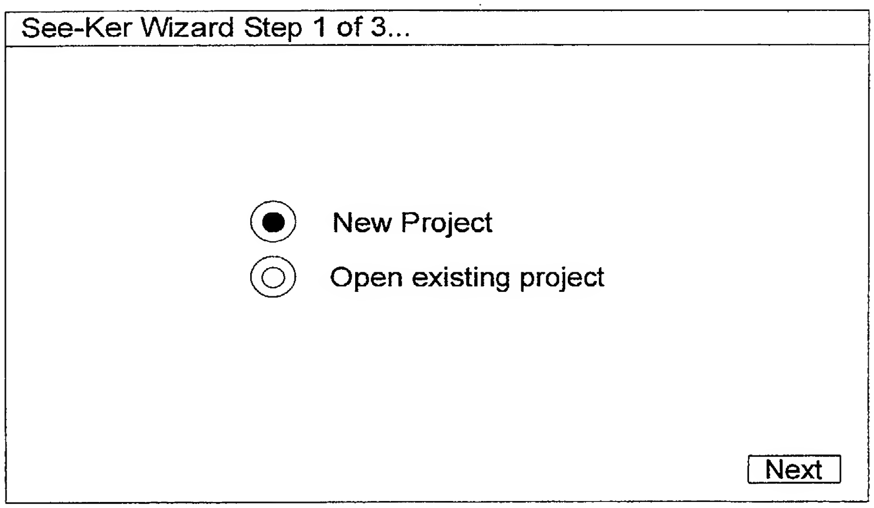

Figure 8

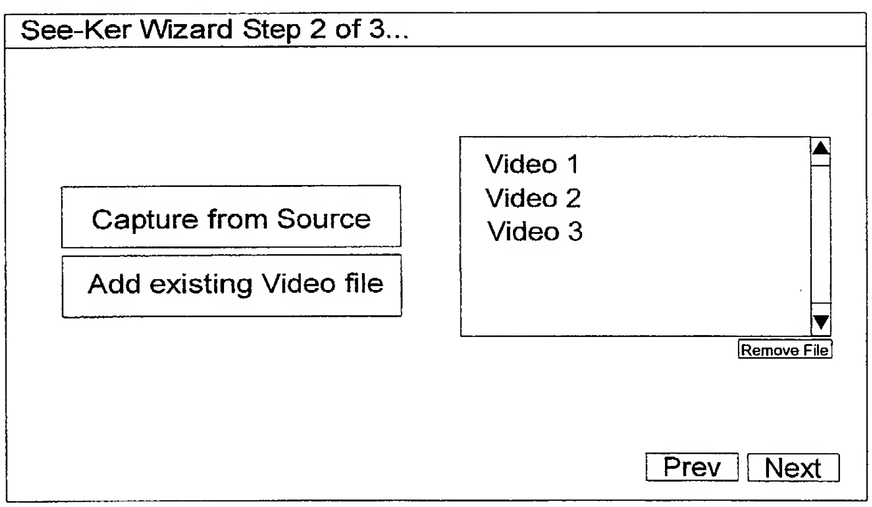

Figure 9

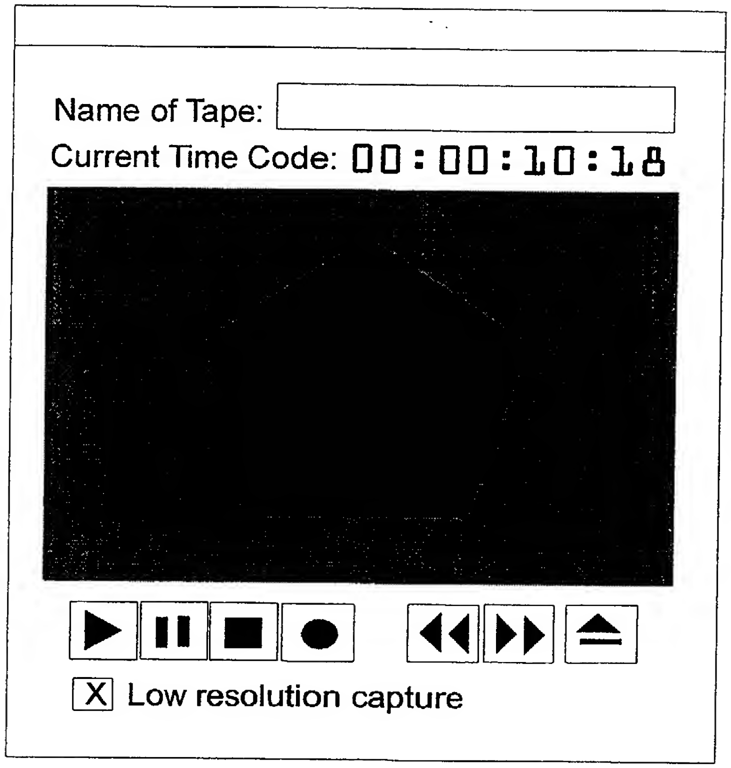
Name of Tape:
Current Time Code: 00:00:10:18
X Low resolution capture

Figure 10

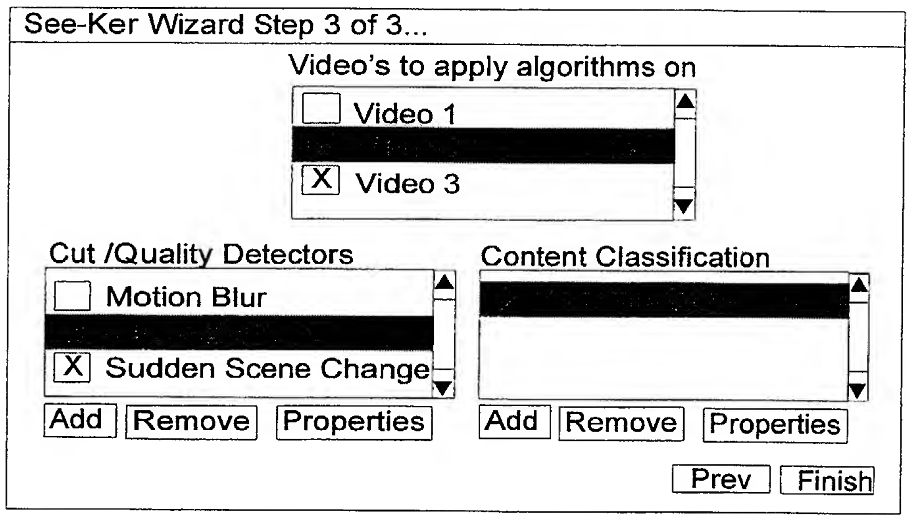
See-Ker Wizard Step 3 of 3...
Video's to apply algorithms on
Video 1
X
Video 3
Cut /Quality Detectors
Motion Blur
X
Sudden Scene Change
Add
Remove
Properties
Content Classification
Add
Remove
Properties
Prev
Finish

Figure 11

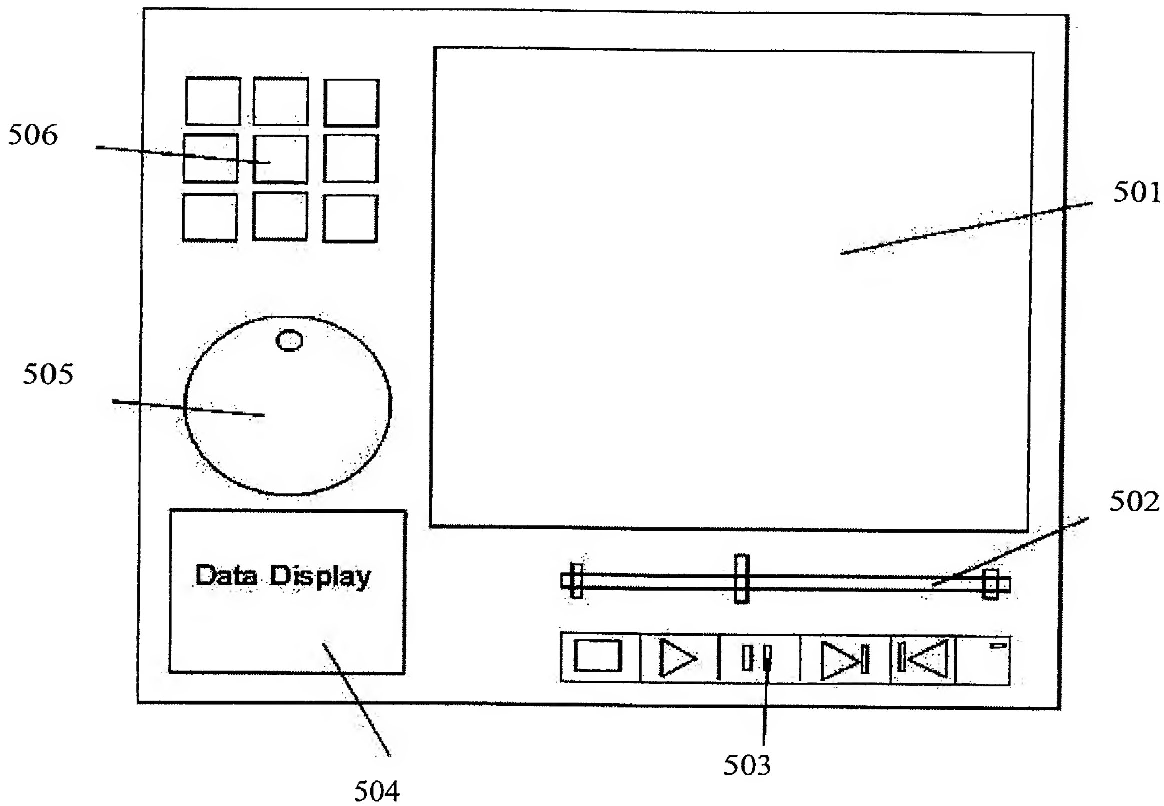

Figure 12

2387710

A Video Editing System and Method

This invention relates to the field of communications and information processing and in particular to the field of video editing.

The advent of digital television and cable/satellite channels means the demand for edited moving image products is seeing a dramatic increase. At the same time, the use of digital video in many fields where traditionally film was used, and its widespread availability to non-professional users, induces many significant changes in the editing and creative processes involved in the making of a moving image product. These changes currently affect quality upstream TV production, domestic and semi-professional camcorder users and internet/www content.

Video editing is, at present, slow and labour intensive. In this, the process of gathering and selecting video material is the main time consuming activity. Currently, selecting video content involves searching, often linearly, through raw video footage, identifying different blocks of footage ("takes") and selecting and extracting into digital form a subset of the blocks. These are then used as building blocks for assembly into the final program. This process is only really manageable for reduced amounts of raw footage. Typically, however, a video production of two hours duration may utilise ten hours of raw footage. Much is therefore redundant (eg. different takes of the same scene) or not usable (e.g. when the camera has been left running during re-

positioning, in between shots, or images are of poor quality). With such a large volume of material, editing becomes a time consuming, laborious and hence expensive process.

Currently there are many video editing packages available. These are aimed at the whole spectrum of home user to professional markets. The packages available include, for example, Adobe (RTM) Dynamic Media Collection; Adobe (RTM) WebCollection; Sony (RTM) Movie Shaker; Apple (RTM) Final Cut Pro; Pinnacle (RTM) Systems: Lightning 500, MicroVideo DV200, StudioMP10; Fast Multimedia DV master pro. These provide an interface and conversion between various data formats. They generally cope with the challenges of editing compressed sequences with limited quality loss, e.g. MPEG motion compensated coding methods. They also enable special effects, e.g. slow motion, colour effects etc. However, known packages do not provide a suitable environment and set of tools to facilitate the material selection and arrangement process. Rather, they assume that this selection has been previously made and focus on later stages of video editing.

With the coming of age of digital video processing, systems that can automatically segment and classify video shots have been proposed. Examples of such systems are described in EP 0915469 A2; EP 0780844A2; US 6,195,458; US 5,635,982 and WO 0103429. Other systems that have the ability to automatically search for specific kinds of shot have been developed. These are targeted at applications such as web browsing or interactive television or video on demand or automatic

editing, as described, for example, in WO 00/39707; WO 99/41684, and EP 1143714A1. A problem with known and proposed systems remains, however, that the basic editing platforms available are difficult to use.

JP200261757 describes a system/process to automatically remove unwanted material and select useful video sections. However, it does not allow for the intervention of the user. This limits the selection criteria to quantitative, computable ones, and hence overrides the creative human input in the selection process.

Various aspects of the invention are defined in the independent claims. Some preferred features are defined in the dependent claims.

According to one aspect of the present invention, there is provided a method for editing digital video footage using a computer processor, the method comprising: cutting the digital video footage into a plurality of clips and sorting the clips into groups based on selected criteria.

By cutting up the video and allocating the resultant clips to groups that can be selected and opened by a user, the editing process is greatly simplified. This is because an editor can very quickly identify relevant clips merely from the groups. For example, the groups may comprise people, landscapes and weather. These simple categories enable the editor to readily locate and put together relevant shots. In contrast, existing packages require editors to scan footage linearly

and pick out clips of interest. Because of the length of typical footage runs, this is labour intensive and so, for the professional, costly.

The method in which the invention is embodied may further involve searching for groups of clips based user input criteria. The method may also involve presenting on a screen or display a group identifier for each group of clips; receiving a user selection of a desired group identifier and presenting clip identifiers for clips included in the selected group in response to the user selection.

The video data may be provided from multiple sources, for instance a video camera, a video tape, files in digital storage media, etc. The video data of each clip may be in different formats.

The step of cutting the video data into clips may be done automatically using a cutting algorithm. As part of this process, the clips can be automatically classified into one or more different categories, relating to the quality of the images or camera motion parameters, for instance. The clips preferably have some uniform characteristics such as photometric characteristics or camera motion characteristics, but are not necessarily semantically significant.

The step of sorting may comprise attributing, interactively or automatically, a set of label descriptors to each clip to facilitate a characteristics-based access and organisation of the video clips. The step of sorting may further comprise a

stage of selecting a group of the video clips (which could be all of them) and organising/sorting them into one or a set of folders.

The method may further involve a stage of automatically outputting and presenting the groups to a user in the form of a set of digital video clips arranged in a hierarchy of digital folders.

The method may comprise presenting the video material gathered and segmented into clips as a virtual representation of the clips, organised according to their categories. Within the categories, the clips may be further organised into classes expressed as labels or attributes. The method may comprise receiving a user-defined classification for a clip and including a clip into a group based on the user defined class.

The method may comprise adding labels/classes to some or all the clips, preferably in response to a user command. The method may involve overriding the labels/classes automatically attributed to some/all of the clips in response to an input received from a user. The method may involve toggling between different categories-based presentations of the clips, so as to use the classes in the different categories as an access criterion. The method may involve applying elementary modifications to the clips, such as merging two clips or splitting a clip in two. The method may further involve modifying the photometric attributes of the images in the clips. The method may comprise creating

hierarchies of virtual folders. The method may further comprise selecting some or all of the clips by virtually copying them into one or more of the virtual folders.

The method may further comprise exporting the data to create automatically a directory structure containing digital video files on a digital storage medium, for example the hard disc of a computer. The structure and virtual content of the video directory as defined in the previous stages of the process is then mapped onto a digital physical folder structure containing physical digital files corresponding to the selected clips.

According to another aspect of the present invention, there is provided a computer program, preferably on a data carrier or a computer readable medium, the computer program being for editing digital video footage using a computer processor, the computer program comprising code or instructions for cutting the digital video footage into a plurality of clips and sorting or organising the clips into groups based on selected criteria.

The computer program may comprise instructions or code for searching for groups of clips based user input criteria. The computer program may also comprise instructions or code for presenting on a screen or display a group identifier for each group of clips; receiving a user selection of a desired group identifier and presenting clip identifiers for clips included in the selected group in response to the user selection.

According to yet another aspect of the present invention, there is provided a system for editing digital video footage using a computer processor, the system comprising means for cutting the digital video footage into a plurality of clips and means for sorting or organising the clips into groups based on selected criteria.

The system may comprise means for searching for groups of clips based user input criteria. The system may also comprise means for presenting on a screen or display a group identifier for each group of clips; means for receiving a user selection of a desired group identifier and means for presenting clip identifiers for clips included in the selected group in response to the user selection.

According to a still further aspect of the present invention, there is provided a digital camera or the like that includes a system or computer program in which the invention is embodied.

According to a yet still further aspect of the present invention, there is provided video content that is produced using the method, computer program or system defined in the other aspects of this invention, or any copies of such video content.

Various aspects of the invention will now be described by way of example only and with reference to the drawings, of which:

Figure 1 is a flow chart of a method for pre-editing video footage;

Figure 2 is a more detailed flow diagram of the method of Figure 1;

Figure 3 is a block diagram of a system for implementing the process of Figures 1 and 2;

Figure 4 shows the degrees of freedom represented by a parameter M, M being used in the step of classifying of Figure 2;

Figure 5 is a representation of three motion parameter domains used in the step of classifying of Figure 2;

Figure 6 is a representation of image regions used in an algorithm for implementing the step of classifying of Figure 2;

Figure 7 shows a main screen of a user interface for implementing the method of Figure 2;

Figure 8 shows a screen that is presented when a user enters the user interface;

Figure 9 is a screen that is presented when the "new project" option is selected from the screen of Figure 8;

Figure 10 is a screen that is presented when the "capture from source" option is selected from the screen of Figure 9;

Figure 11 is a screen that is presented to allow the user to vary cutting or classification criteria, and

Figure 12 is a screen that is presented to allow users to select cutting points, when trimming a clip.

Figure 1 shows the steps in a method for editing video footage. These include receiving raw data from a video input; dividing the raw data into smaller video clips and sorting or allocating the clips into groups or sets based on

selected criteria. A more detailed flow diagram is shown in Figure 2, in which the step of allocating the clips to groups involves various stages of selection, classification and sorting. This will be discussed in more detail later.

The methods of Figures 1 and 2 can be implemented using the arrangement of Figure 3. This has an interface 201 to the video data, which data may be obtained from different data input sources 200, such as a video camera, a video tape, files in digital storage media, etc. To deal with data in analogue form the interface may include a frame grabber. It could also have a firewire port if the data is on tape. Additionally or alternatively, a connection to a hard drive could be provided, if the data is in digital form.

The video interface 201 of Figure 3 is able to communicate with user interface software 203 that is run by a processing unit 202. The user interface software 203 is a computer program, which is adapted to implement the method of Figures 1 and 2. Clips and other video content used or created by the interface software 201 are stored in a digital storage unit 204. As will be appreciated, the video editing platform of Figure 3 could be implemented in a standard PC or any such device or product with similar processing capability. More specifically, the platform could be implemented in a digital camera, with the video content being provided by the camera, or any other such consumer electronics device.

Returning to Figure 2, this shows various different steps of the process, starting from a data import stage when video

footage is introduced to an export stage when selected and sorted digital video clips are output. Any path outlined by the arrows in Figure 2 is possible between the import and export stage. The user interface software is adapted to allow the steps of Figure 2 to be tackled iteratively in a non-linear way, going back and forth between the different parts of the process as many times as required. However, for ease of explanation, each step of the process is described successively, in a linear way.

The first stage in the method of Figure 2 consists of assembling the raw video material that is to be edited. This can be in the form of analogue or digital video format and can be provided on any storage medium, for example tape, disc, memory, etc. This data is then digitised where necessary, and put in the form of a windows supported video format. For analogue data the step of digitising can be done using an analogue frame grabber. The result of this stage is a representation of the raw video material as a set of digital video files on a digital storage medium. The files may be compressed or have a low resolution. These files constitute the basis of the representation of the raw video material for the overall process.

During the import stage, enough information is kept in order to be able to re-iterate the import for all the used sources, producing the highest possible resolution for each of the digital files. The software is adapted so that all of the raw video material to be used for the final edited product can be imported at the start of the process, or alternatively

new raw data can be added as needed at any stage of the process.

Once the video content is digitised, the files are then divided or cut into a series of clips or sets of successive images that present certain homogeneity in one or more low-level image characteristics. This can be done using a histogram or statistical analysis of the data or by assessing the textural characteristics of the image or by looking at movement patterns, such as camera motion between frames. Cutting may be done automatically using a cutting algorithm included in the user interface software. In a preferred embodiment of the invention, two algorithms are used to perform cut detection. The first of these algorithms detects cuts by comparing histograms of intensity levels between frames; the second partitions the video by detecting the appearance of intensity edges that are distant from edges in the previous frame.

Consider firstly the histogram algorithm. This compares the grey level histograms of consecutive frames $D(i, i+1)$. A cut is declared if the absolute sum of histogram differences between frames is greater than a threshold T,

$$D(i,i+1)=\sum_{j=1}^{n}\left|H_i(j)-H_{i+1}(j)\right|$$

where $H_i(j)$ is the histogram value for the grey level j in the frame i, j is the grey value and n is the total number of grey levels (256 in the current implementation).

The principle behind the histogram algorithm is that two frames having an unchanging background and unchanging objects show little difference in their respective histograms. Although there may be cases in which two images have similar histograms but different contents, the probability of such an event occurring in practice is sufficiently low that such errors can be tolerated. This is discussed in "Automatic Partitioning of Full-Motion Video" by Zhang et al, Multimedia Systems, 1993, pp. 10-28.

As an alternative to the histogram algorithm, an intensity edge algorithm can be used to determine where cuts in the video should occur. An example of this method is described in detail in the article "A Feature Based Algorithm for Detecting and Classifying Production Effects" by Zabih et al, Multimedia Systems 1999, pp. 119-128. The intensity edge method involves comparing the fraction ρ of edges entering or exiting between two frames: $\rho = \max(\rho_{in}, \rho_{out})$, where ρ_{in} measures the proportion of entering edge pixels and ρ_{out} measures the proportion of exiting edge pixels. This represents the fraction of changed edges. This approach is based on the assumption that during a cut or a dissolve, new intensity edges appear far from the location of old edges. In a similar way, old edges disappear far from the location of new edges. Hence, by counting entering and exiting edge pixels, cuts can be detected.

The intensity edge algorithm takes two consecutive frames I and I' and firstly transforms colour images to grey images. Edge detection is then performed using the classic Canny algorithm. This algorithm is described in "Machine Vision" by R Jain et al, Mc Graw Hill International Editions, Computer Science Series, 1995. The Canny type edge detection produces binary images E and E', in which edges are denoted by white pixels). Next, copies of E and E' are created with the pixels dilated by a radius r, to form expanded images D and D'. Then the fractions of entering and exiting edge pixels are calculated.

Consider ρ_{out} the fraction of edge pixels in E which are farther than r away from an edge pixel in E'. A white pixel $E[x,y]$ is an exiting pixel when $D'[x,y]$ is not a white pixel. This is because the white pixels in D' are exactly those pixels within distance r of an edge pixel in E'. The equation for ρ_{out} is:

$$\rho_{out} = 1 - \frac{\sum_{x,y} E[x,y]D'[x,y]}{\sum_{x,y} E[x,y]}$$

This is the fraction of edge pixels that are exiting. ρ_{in} is calculated similarly:

$$\rho_{in} = 1 - \frac{\sum_{x,y} D[x,y]E'[x,y]}{\sum_{x,y} E'[x,y]}$$

Scene breaks are detected by looking for peaks in the edge change fraction ρ. When such peaks are identified, the video content can be divided. In this way, a stream of video can be divided into a plurality of clips, each presenting within itself a certain degree of homogeneity.

Whilst histogram and intensity edge detection methods are described above, the process of dividing video into clips can be realized automatically, using various methods and algorithms. Indeed, as image processing algorithms become more sophisticated, semantically significant characteristics of the image could be used for the division of the material into clips. For example, algorithms could be used to detect the presence/absence of a talking head or a certain type of object/creature or location-based characterisation or project-specific characteristics. Some suitable cut detection methods are presented in US 6,195,458, US 5,635,982 and US 5,339,166. Still further methods are described in the articles "Automatic partitioning of full-motion video" by Zhang et al, Multimedia Systems Vol. 1, 1993, pp. 10-28; "Temporal Video Segmentation: A Survey" by Koprinska et al, Signal Processing: Image Communication Vol. 16, 2001, pp. 477-500 and "A feature-based algorithm for detecting and classifying production effects" by Zabih et al, Multimedia Systems Vol. 7, 1999, pp. 119-128.

Once the video is divided, the system software is adapted to extract an internal representation of the clips as a pointer to a time index in one of the files extracted from the import phase, along with a duration or clip length. The

representation and display of the clips in the other stages of the process is based on these pointers. The clip pointers or "virtual clips" are the basic units that further stages of the process manipulate. The end product of the process is a set of real digital clips corresponding to a subset of these digital clips, that is, a set of video files each containing at least one of the clips.

In order to introduce flexibility and creativity into the cutting process, the user interface software is adapted to allow the automatic division into clips to be overridden by the end user. For example, the software may allow the user to split an existing clip into two or merge adjacent clips together. This means the user can choose to divide the data into clips according to higher level criteria such as semantic content. These video chunks are often referred to as "scenes".

The process of Figure 2 further builds on the representation of the clips by attributing of a series of labels or classes to each clip. The clips can be classified in various ways, for example, according to clip start time and clip duration. This classification can be done automatically. Other classifications fall into a set of broad categories such as camera motion type; image quality criteria; user-defined classes, etc. In this case, the process of classifying is either done automatically into a set of pre-defined classes through the use of image processing classification algorithms or manually by the end user into a set of user classes that are defined dynamically. To allow for manual intervention,

the user interface software is adapted to allow the user to override the classification output by the algorithms. In this way, the user can change any of the class labels of the clips as and when desired.

The classification algorithms could characterise the camera motion over the clip or the image quality, e.g. colour balance, contrast, blur/focus, etc. Algorithms for detecting scene changes and camera motion are known. Various examples are described in the articles "Efficient Scene Change Detection and Camera Motion Annotation for Video Classification", by W. Xiong and J. C-M. Lee, Computer Vision and Image Understanding Vol. 71, PP 166-181; "A Unified Approach to Shot Change Detection and Camera Motion Characterisation", by P. Bouthemy, M. Gelgon and F. Ganansia, IEEE Transactions on Circuits and Systems for Video Technology, Volume 9, No. 7, October 1999 and "The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields" by M.J. Black and P. Anandan, Computer Vision and Image Understanding, Volume 63, No. 1, January 1996, pp. 75-104; "Out-of-focus blur estimation and restoration for digital auto-focusing system" by Kim et al, Electronic Letters, 1998, v. 34, n. 12, June. As regards blur detection, a suitable algorithm for implementing this is described in the article "Focusing" by E. Krotkov,International Journal of computer Vision, 1987,v. 1,pp. 223-237.

Whilst many different classifications are known and could be used, a preferred option is to classify all of the video

clips into a motion label. This could, for example, be panning, tilt, rotation, zoom or even static. It could also be unrecognized, where the camera motion is chaotic, or complex, where the camera motion is defined by two or more simple motions.

The preferred motion classification method uses Bouthemy's motion estimation. This estimates the motion between two consecutive frames, computing the values of a parametric or affine model (a_1, a_2, a_3, a_4, a_5, a_6). Then, a new model M is defined from the previous one (resulting from a linear combination). In this case, $M=(m_1, m_2, m_3, m_4, m_5, m_6)$, where $m_i = \{0, >0, <0\}$, $i \in (1 \ldots 6)$ and each of these parameters represents an easy physically meaningful interpretation of the dominant motion: m_1 - horizontal translation; m_2 - forwards or backwards translation; m_4 - vertical translation; m_5 - rotational motion and m_3, m_6 - hyperbolic terms. Figure 4 shows a graphical representation of these parameters.

At each instant t, the method interprets the value of the vector M and converts it into a motion label. This operation has to be repeated over all the frames of the clip so as to determine the global motion. Due to noise, estimation errors, and the use of an approximate motion model, the components of the vector M cannot be exactly equal to 0, even if it should be the case. The aim is then to decide whether these estimated values are significant or not. To do this, the following likelihood log-ratio test is performed:

$$r_i = \ln\left(\frac{f_0(m_i)}{f_1(m_i)}\right) \underset{H_0}{\overset{H_1}{\lessgtr}} \lambda \quad \text{with } i \in (1 \ldots 6)$$

where $f_0(m_i)$ is the likelihood function for $m_i = 0$ and $f_1(m_i)$ the likelihood function for $m_i \neq 0$. If the ratio is lower than the threshold λ, the component m_i is declared to be significant, otherwise it is considered to be null.

The threshold λ is determined by the value of $\min(r_i)$. In fact, it is assumed that the most relevant parameter is the one that has a minimum log-ratio. Hence, parameters having a log-ratio less than a percentage of the minimum log-ratio are not significant. For the purposes of the present method, 5% is deemed to be a good compromise. Hence, the threshold λ is defined as 5% of the value of $\min(r_i)$.

The above estimation algorithm is relatively slow, i.e. not real-time. Because of this, a small frame resolution (160*120) is used and the estimation parameters are set to the smallest value. In this way, the processing time can be speeded up. However, a disadvantage of this is that the significance of one or many parameters modifying the type of global motion may be artificially increased, thereby making some otherwise insignificant parameters seem dominant. This issue can be partially addressed by using a second threshold γ to enable the most dominant parameters to be distinguished from parameters that are less dominant, but not insignificant. These less dominant parameters are referred

to as being influential parameters and are defined as having a log-ratio of between 5 and 20% of $\min(r_i)$. The boundaries between the dominant, influential and insignificant parameters are shown graphically in Figure 5.

Although not as significant as the dominant parameters, the influential parameters themselves may include some real information as well as information that may have been produced erroneously. For example, in a video clip where something is moving around (like people in a store), the influential parameters describe this type of secondary motion. Hence, influential parameters actually provide information about the scene and what happens inside it. So, in some cases, it would be useful to interpret these parameters. The difficulty is however differentiating the relevant information in the influential parameters from the noise. To do this a parameter representation model is used.

The parameter representation model uses six parameters (Model M) which can be positive, negative or null. Now, a new model is defined to represent them in a binary system. This is a very simple way to store information on each parameter. The model is defined as follows:

$$\psi=(\psi_1,\ \psi_2,\ \psi_3,\ \psi_4,\ \psi_5,\ \psi_6,\ \psi_7,\ \psi_8,\ \psi_9,\ \psi_{10},\ \psi_{11},\ \psi_{12})$$

with $\psi_i = \{0,\ >0\}$, $I\in(1 \ldots 6)$ and $\psi_i = \{0,\ <0\}$, $I\in(7 \ldots 12)$. Using this, a panning left is characterized by $(0,0,0,0,0,0,1,0,0,0,0,0)$, whereas a panning right is $(1,0,0,0,0,0,0,0,0,0,0,0)$.

In the present case, two models are created. The first one represents the dominant parameters and the second one represents the influential parameters associated with the dominant parameters. Each of the two provides a vector, one describing the dominant parameters and another representing the influential parameters. To determine the dominant motion between two frames, only the dominant parameter vector is used. This gives an indication of the global motion. This step is repeated for all the frames of video in order to obtain a global picture of the motion.

To determine the motion within a video clip, information on every second frame is stored. This information is dominant motion parameter, influential motion parameter, and time parameter. The last parameter gives the time position of the frame within the clip in seconds. To store this information a list of records is created, this list representing the dominant motions appearing in the video clip. Each record has a counter that is incremented by a unit every time information obtained matches with one of the records created. To determine whether matches exist, time motion information from two consecutive frames is obtained and both the dominant motion and time parameters are compared to those already existing from the list record. A match is defined as occurring when the dominant motion parameter is the same as the one from the record and the time parameter is less than ten times the record time parameter. In the event that there is a match, the relevant counter is incremented by one unit and the existing time parameter is refreshed or up-dated with

the new time parameter. If there is no match, a new record is created for the motion at the end of the list. Once this is done for all of the frames of the clip, if the list yields a motion appearing in more than 50% of the number of video frames, then this motion is considered as the dominant one. Otherwise, it is assumed that the video clip has no global motion.

Another method of classifying video clips uses optical flow techniques. To compute the optical flow, the very well known Lucas and Kanade's algorithm is used. This is described in detail in the textbook "Introductory Techniques for 3D computer vision" Emmanuele Trucco, Alessandro Verri Prentice Hall. A basic assumption in computing optical flow is that the brightness of every point of a moving or static object does not change over time. In order to determine optical flow, each frame of the video is divided into four regions, as shown in Figure 6. Once this is done, the number of pixels not moving is counted. If more than 20% of the pixels are moving then it is assumed that there is camera motion on the scene. If more than 80% of the pixels are not moving then it is assumed that there is no camera motion on the scene. If there is camera motion, the direction of the optical flow is analysed by calculating the mean and standard deviation of the motion vectors. If the standard deviation is smaller than a fixed threshold τ then it is either a pan or a tilt and the direction is determined by the mean direction of the optical flow vectors. Otherwise, if the standard deviation is bigger than τ, the motion is either a zoom or a rotation and the mean in the regions is used to determine if it is a zoom

out/in or the direction of the rotation. The idea behind this is that on a pan or tilt most of the pixels are moving in one direction, thus the standard deviation is small. In contrast, in a zoom or rotation, pixels are moving in different directions and therefore the standard deviation is bigger.

After finding the type of motion, the average direction of the optical flow is used to determine the direction of the motion. In the case of pan and tilt, the average direction of the optical flow in the whole image is used. This is because most of the pixels should be moving in the same direction. For the zoom and tilt, the average direction of the optical flow is compared in different regions according to observed patterns. For example, in the case of a zoom out $\theta_{IV} > \theta_{III} > \theta_{II} > \theta_{I}$, $\theta_i \in (-\pi, \pi)$, where θ_i is the average direction of the optical flow on region i. It should be noted that the condition $\theta_{IV} > \theta_{I}$ is unique for the zoom out.

θ_i and σ are the average of the direction of the optical flow on region i and standard deviation of the direction of the optical flow on region i respectively. When no index is specified they denote the average and standard deviation in the whole image, respectively. The classification criteria are then set so that if $\sigma > \tau$ then the motion is a pan or tilt. In this case, θ is used to determine the direction of the motion. Otherwise, camera motion is determined to be a zoom or rotation. More specifically, if $\theta_{IV} > \theta_{I}$, then it is a zoom out; if $\theta_{II} > \theta_{IV}$, then it is a zoom in; if $\theta_{III} > \theta_{IV}$,

then it is a rotation anti-clockwise and if $\theta_I > \theta_{II}$, then it is a rotation clockwise. In the current implementation, every nineth frame is analysed, although, the software may be operable to allow the user to vary this. To consider a scene to constitute a category or class at least 50% of the analysed frames must be of the same category, otherwise it would be considered a 'Complex Motion'.

In addition to motion detection, the clips can be sorted according to whether or not they are blurred or the level of contrast. As regards blurring, this can be detected using various different techniques. As a preferred option, blurring is detected using Canny's edge detection algorithm. As mentioned previously the Canny algorithm is very well known. The blur detection is based on the principle that if the detected edges are not well defined then the image is blurred and if the edges are well defined the image is not blurred. Experimentally, it was found this is a useful measure of how to predict levels of image blurring.

Contrast detection can also be used as a means for classifying clips. This can be useful because video sequences often include frames that are white or black, indicating that there has been a cut or change of scene. In some other cases detail cannot be detected, because frames are too bright or too dark. The reason of lack of contrast, in other words the existence of image frames that are "too bright" or "too dark", is that pixels within the image that represent the objects or part of objects tend to have intensity values which are very similar/close to each other.

This can be observed in a histogram of the intensity values of the image. The histogram of a poorly contrasted image is very narrow, in other words the pixel values are clustered in a narrow area of the histogram, while the histogram of an image with better levels of contrast would be more spread out.

The histogram of an image is a discrete function that is formed by counting the number of pixels in the image that have certain intensity values. When this function is normalised to sum up to 1 for all the intensity level values, it can be treated as a probability density function that expresses the probability of a certain intensity value being found in the image. Seen this way, the intensity value of a pixel is a random variable that takes values according to the outcome of an underlying random experiment. Ideally, a well-equalised image would have a flat-ish histogram. In theory, it would then be possible to enhance an image to have a relatively flat histogram, so that the pixels can be randomly re-distributed across neighbouring intensity values.

Since the aim here is to detect and classify the lack of contrast in an image, looking at the intensity values of the pixels as a random variable is useful. As mentioned before, a good distribution of pixel intensity values is indicative of a well-contrasted image. Hence, this distribution could be measured through the standard deviation of the intensity levels. If the standard deviation is high, a wide distribution of the intensity pixel values is present,

implying good contrast. Otherwise the image has poor contrast levels.

In order to distinguish between good and poor levels of contrast, thresholds are set. A flat histogram was assumed as an example of a well-contrasted image in order to set the standard deviation threshold. The histogram was divided in three areas, corresponding to the first 20%, the next 60% and the last 20% of the bins. The first 20% of the bins are associated with dark intensity values, the next 60% are indicative of medium intensity values and last 20% are bright intensity values. Experimentally it was found that if the standard deviation of the intensity values is less than 4, a black or a white frame is present. To discriminate between black and white, the intensity values are checked to determine the area of the histogram they belong. If more than 50% of the intensity values are in what was defined the dark area, a dark frame is present. If more than 50% of the intensity values are in the bright area of the histogram, a bright frame is present. In this way frames that are "too dark" or "too bright" can be detected.

Whilst the methods described above are preferred, the process of classifying the clips can be realized automatically, using various methods and algorithms. For example, the clips could be classified on the basis of semantically significant criteria or whether or not there is a "talking head" in the clip.

Once the clips are classified according to whichever classification method is used, a sorting process is conducted. Based on the class labels attributed to the clips, the user interface software automatically presents the clips to the end user in an organised form. Typically, the clips are arranged in an array of folders, each folder representing a particular classification. The end user is then able to repeatedly select amongst different sorting criteria, which correspond to the previously mentioned categories, such as: camera motion type; image quality criteria; user-defined classes; clip start time; clip duration, etc. The clips are then presented in an organised way where all clips with the same label within one category are grouped together. This automatic sorting facilitates characteristics-based access to the material for the selection.

Having cut, classified and sorted the clips, the video can be readily selected for use. This aspect of the process is realised interactively by the end user. The present invention facilitates this stage through a suitable virtual representation of the video clips and of the storage structure in which they are to be organized, and through enabling a suitable level of interactivity. The virtual representations are presented in groups, according to the clip classifications, via a software generated user interface. This will be described in more detail later. The user can then select some or all of the virtual clips and arrange them into a hierarchy of folders that he defines himself.

The last stage of the process involves exporting the clips selected by the user. To do this, digital files are created on a digital storage medium, organised in a real directory structure so as to map the virtual representation of the video clips and of the storage structure established during the previous stages of the process. The video files are extracted from the already digitised, possibly low-resolution version of the video data used as a basis for the clip segmentation and classification in the previous stages. Alternatively, they can be extracted from the original video sources, that would then where relevant be recaptured at the requested resolution. In this way, the method provides edited video content as selected by the user. Of course, this video content could be processed further to improve quality and/or copied to provide multiple copies for distribution or otherwise.

In order to allow a user to edit video and/or access pre-existing video clips, a user video interface is provided. The interface software is aimed at taking the user through the steps required to add video into the system in a rapid and friendly manner. The main interface screen is shown in Figure 7. However, the software is operable to present a wizard upon entry into the system and/or when a new project is being created. The first wizard screen is shown in Figure 8 and is titled "Wizard Step 1 of 3". This provides the user with two selectable options, the first being to start a new project and the second being to open an existing project.

If the existing project option is selected and the "next" button clicked, the screen of Figure 7 is presented to the user. This will be discussed in more detail later. If the new project option is selected and the next button is clicked, the screen of Figure 9, titled "Wizard 2 of 3", is presented. This allows two further options, firstly the capture of video from a source and secondly the addition of an existing video file. If the capture from source option is selected, a list of all available Windows video capture devices is presented. When the Firewire (IEE-1394, iLink) device is selected then various options to remotely control the device are presented. Once the relevant device is selected, the "next" button is clicked and the user is presented with the screen shown in Figure 10.

Figure 10 shows the selected clip in the main view screen and provides various remote control options that are selectable via a series of interactive buttons. These buttons include play, pause, record, stop, rewind, fast-forward and eject (if available). Figure 10 also shows a data entry field for allowing the name of the tape or source medium to be input. This sets up an explorer style clip folder structure. If the eject option is available then it resets the name of the tape when ejected. Also provided is an option to capture at lower resolution (half) and run all the editing/clip selection on that and then batch capture all the necessary clips in full resolution at the output stage. If video is captured at low resolution, the software remembers this so at the output stage the video may be automatically recaptured at full

resolution by the software.

Selection of the record button of Figure 10 causes an interactive "save dialogue" pop-up box to be displayed (not shown). This "save dialog" box allows the user to select where to store the captured file and what name to give to the file. There is a default file name, which is used in the event that the user does not provide an alternative. The dialog remembers what directory it was last initiated in so that on future instances the user does not have to navigate back to the directory. Once the save parameters are entered by the user and received by the software, the "save dialogue" box is exited and the software automatically triggers the recording of the video content on the digital storage unit 204 of the system of Figure 3. This recording is done at the selected resolution for the device (camera, VCR etc). After the video has been captured the selected file name is added to a listbox in the screen of Figure 9, e.g. as files video 1, video 2, video 3 etc.

Returning now to Figure 9, in the event that the "Add video from existing files" option is selected, the standard Windows open file dialog is presented. After the "Ok" button is pressed in the open file dialog, the selected file is added to the listbox shown in Figure 9. A file from the listbox can be removed if it has been added by accident.

Once the desired video is selected using the screen of Figure 9, the next button is selected. This causes the screen of Figure 11, titled "Wizard Step 3 of 3" to be presented. This

allows certain aspects of the functionality to be customised by the user, such as the type of cut detection or classification required. Algorithms are integrated into the software as plug-ins, and therefore the set of available classifiers and cut detectors can be upgraded or modified very easily.

Figure 11 has three selection fields. The first allows the selection of desired video footage from the list of video files that were chosen by the user when in the screen of Figure 9. Another of the fields lists the basis on which the selected video is to be cut, for example on the detection of motion blur; focus blur and sudden scene change. Selection of one or more of these options determines the algorithms that are to be used to carry out the stage of cutting the video footage into clips. As described with reference to Figures 1 and 2, the cut detection algorithms could use a histogram or an intensity edge approach. However, many different algorithms could be added either to qualify the image quality (color balance, brightness…) or the image content at a lower or higher level of understanding ("moving objects in image", "head and shoulder", "landscape"…).

The last data entry field of Figure 11 presents various options for the step of classifying the clips. In the screen shown, the option "pan direction" is selected. Other options such as motion type or image quality as previously described could of course be presented to the user for selection. In this way, the user is able to choose a particular type of clip classification as desired.

When the user has made all of the relevant selections, the "finish" button is clicked. The user selections are then received by the interface software and interpreted and the relevant algorithms are run. The cut algorithm is run first to divide the video into clips and then the classification algorithms are run to act on the clips obtained after completion of the cutting stage. To give a visual indication to the user that work is being done, a progress bar is provided. After the step of classifying is completed, the clips can be sorted either automatically or in response to a user selection. The groups can then be presented to the user for selection and viewing. The main display of Figure 7 is used to allow users to select sort criteria, present clips and select those clips for export.

Figure 7 shows the main system display. This has a file menu 301; a preview window 303; a sort selection window 302; a timeline 304; a clip labeller 308; a clip icon display 310; a trash can 311 and a clip explorer 312-313.

The file menu 301 has various options that are displayed when the "File" button is selected. These options include new; open; save; save as; add new clips; export and exit. The "New" option clears out the current project and starts a new one. The "Open" option clears out the current project and shows the standard Windows open file dialog. "Save" saves the current project using purpose built format. If the project has not been saved before then it will be saved. "Save as" shows the standard Windows save dialog box and if

selected saves the project using the using the purpose-built format. "Add new clips" displays the wizard of Figure 9 so that the user can go through the previously described steps for adding content. It should be noted, however, that in this case, the wizard title is changed to "... Step 1 of 2" and the "previous" button is removed. "Export" takes the virtual explorer folders and replicates that structure on disk. If the clips have been captured at low resolution then this provides the option of batch capturing at full resolution. "Exit" exits the application. If the project is unsaved or changed then a prompt is generated to ask if the user wants to save the project.

The preview window 303 is the gateway between the main display and the video viewer/editor window where adjustment of cut location may be done. Clips can be dragged from the timeline 304, or the clip explorer 312-313 or the clip sorter 309 to this window. They are then automatically played in a loop. After the first loop has been completed the video is returned to the start and paused. When the user clicks once on this window the player alternates between pausing and playing the video. Double clicking opens up the video viewer/editor window. This is shown in Figure 12.

The screen of Figure 12 has a main display 501 for viewing video clips and a bar 502 with a selectively movable marker that can be used to select the cutting points in the video. Movement of the bar marker causes movement of the image on screen. To select the start of the desired clip the software is operable to respond to receipt of a signal generated when

the left button on the user's mouse is clicked. When such a signal is received the start position is set at the location corresponding to the position of the marker bar. To select the end of the desired clip, the software is operable to respond to receipt of a signal generated when the right button on the user's mouse is clicked. In this case, when a right click signal is received the stop position is set at the location corresponding to the position of the marker bar. In this way the length of the clip can be defined. Also provided on the screen of Figure 12 are a browse bar 503 and a jog control 505 that allow the user to browse through the video footage. As these controls are varied, the video displayed on display 501 changes accordingly. Function buttons 506 are optionally provided to allow the user to manipulate the image. These buttons could, for example, allow the contrast to be enhanced or image balance corrected.

Returning now to Figure 7, the sort selection window 302 provides various options to allow a user to select the type of sort that is to be used to display all the data in the clip sorter 310. For example, the clips can be sorted by time. In the event that this is the user selection, the software searches through the video clips and groups them according to time section, which time section may or may not be determined by the user. In this case, each tab sheet 309 in the clip sorter 310 then corresponds to the selected time sections of the data. If there are multiple video files then each tab sheet in the clip sorter 309 is the clip name and each one of these tabs has sub-tabs divided as before.

The clips can also be sorted according to the automatic classifications. This can be done using the class labels attributed to the clips in the earlier classification stage. If the automatic option is selected, the user interface software automatically looks for the default classifications, searches for clips having class labels associated with these and presents the clips found to the end user in an organised form. Typically, the clips are arranged in an array of folders, each folder representing a particular classification such as: camera motion type; image quality criteria; contrast levels etc. The clips are then presented in an organised way where all clips with the same label within one category are grouped together. In this case, each tab sheet in the clip sorter 309 corresponds to one of the categories that is output by the automatic classifiers. The categories could include "Fixed shots", "Panning", "Zooming", "Other", etc. Sorting may also be done by image quality. For example, each tab sheet in the clip sorter 309 may correspond to categories such as "Dark", "Light", "Poor contrast", "Blurred", "shaky", "OK". Further sorting criteria may be based on clip/shot duration or user categories. In the case of sorting by duration, each tab sheet in the clip sorter corresponds to ranges of duration for the clips, e.g.: "from 1 to 2 minutes".

For sorting by user categories, the software is adapted to receive and store user attributed classifications associated with the clips, which user classifications are manually entered. To allow this, the software is operable to allow users to customise their own category list, e.g. "Landscape",

"Static Landscape", "People", "Long shot", "Close-up". Visual inspection of the clips allows users to allocate specific clips to their own customised categories. Of course each clip may be allocated to a plurality of different categories.

Other sorting options are possible. For example window 302 could include an image quality option. In this case, the interface software would search for all clips that have been classified with an image quality label and group the clips together accordingly. Of course, as will be appreciated, in this case the clips that are found could have been automatically or manually classified by the user. As before, each tab sheet in the clip sorter would then correspond to categories such as "Dark", "Light", "Poor contrast", "Blurred", "shaky", "OK".

The timeline 304 shows the cut location for the clips relative to their position in the video. In this, selected clips are highlighted and below the clip a colour-coded classification 306 is provided, this being based on the automatic classification. In the bottom right of the timeline is provided an indication of whether the clip has been moved to the clip-explorer 313. Clips that are deemed unsuitable are hashed over. This is also done to clips that have been placed in the trash-can 311.

Clips in the timeline 304 or the clip icon display 310 can be "dragged and dropped" to the preview window 303 or Trash Can 311. Single clicking on a clip highlights it and shows it in

the preview window 303, thereby allowing the user to watch at least a segment of the clip. Single clicking also causes the clip data to be shown in the clip labeller 308. In addition, a go to tab and clip within that tab is shown in the clip sorter 309-310. If a clip is selected from the timeline 304 and the shift key is held down and an adjacent clip is clicked on, then the two clips are selected and the merge clip button 307 is enabled. Clicking on this button 307 merges the clips together. The data of the first clip in time order is then used as the new clip data.

The clip icon display 310 should be the same as in the timeline 304 with the addition of clip info below the icon. The clip sorter displays all the clips according to the sorting criteria selected in the type of sort 302. It is assumed that the end user is familiar with the raw material and will be using this feature for rapid shot selection, e.g. "I need a static landscape shot". The clips can then be sorted by automatic classification or if they have labelled the shot they can sort by user categories. The clip sorter supports drag-n-drop to the clip explorer 312-313 or preview window 303 or trash can 311. Single clicking on a clip icon highlights it and shows it in the preview window 303. This also causes the clip data to be shown in the clip labeller 308 and highlights the clip in the timeline 304.

The clip explorer 312-313 provides a Windows explorer presentation of clips to enable user to create a virtual clip folder hierarchy. It also provides the ability to delete named virtual folders and to drag-n-drop clips from clip view

to trash can 311 and to preview window 303. Single clicking on a clip highlights it and shows it in the preview window 303. This also shows the clip data in the clip labeller 308 and causes the clip to be displayed and highlighted in the timeline 304 and clip sorter 310.

To enable files to be added to the clip explorer 312-313, the user interface software is operable to allow each of the sorted clips 305 shown in the timeline 304 or the clip icon display 310 to be selected by the user. This is done by allowing the clip icons to be "dragged-and-dropped" into the clip explorer 312-313. Each selected file is added to a virtual file of selected clips. The order of the clips can be determined be the user. When all the desired clips are chosen and added to a virtual folder, the "Export" option in the file menu is selected. This takes the virtual file and replicates that structure on disk. This causes the automatic creation of a video directory structure for containing digital video files. This can be stored on, for example the hard disc of the user's computer. The structure and virtual content of the video directory is then mapped onto a digital physical folder structure and the physical digital files corresponding to the selected clips are retrieved and stored in the physical folder structure. At this stage, if the clips were captured at low resolution then the Export option allows batch capturing of the selected clips at full resolution. In this way, edited video content is provided.

The user interface of Figures 7 to 12 provides a simple and effective interactive platform for editing video content in

accordance with the methods of Figures 1 and 2. As will be appreciated, various data inputs and video file formats could be used. For example, DV Video and other supported windows video capture devices; MPEG; MJPEG; other Windows AVI supported file formats and previously processed video content. Once the content is received it is divided into clips. The step of dividing can be done automatically on the basis of default criteria. However, to improve flexibility and functionality for the user, the interface software is operable to interactively receive criteria selected by the user and use these to determine where the video should be cut. In the event that trimming of the clips has to be done or the user wishes to over-ride the automatic cutting, an editing screen is provided to allow manual manipulation of the cutting process. The software is also operable to classify the clips. Again, the classification can be done on the basis of default or user determined criteria, such as camera motion or blurring etc. Once the video is cut and classified, the clips are sorted into groups based either on the selected classifications or on other user entered criteria. Each group of clips can be presented on screen, so that a user can very quickly and easily find what he is looking for. In this way, it can be seen that the platform in which the present invention is embodied provides a simple, interactive and fast mechanism for editing video content.

A skilled person will appreciate that variations of the disclosed arrangements are possible without departing form the invention. Accordingly, the above description of a specific embodiment is made by way of example only and not

for the purposes of limitation. It will be clear to the skilled person that minor modifications may be made without significant changes to the operation described above.

Claims

1. A system for editing digital video footage using a computer processor, the system comprising cutting means for cutting the digital video footage into a plurality of clips and sorting means for sorting the clips into groups.

2. A system as claimed in claim 1, comprising means for presenting on a screen or display a group identifier for each group of clips; means for receiving a user selection of a desired group identifier and means for presenting clip identifiers for clips or clips included in the selected group in response to the user selection.

3. A system as claimed in claim 1 or clam 2 comprising means for searching for groups of clips based on user input criteria.

4. A system as claimed in any one of the preceding claims, wherein the means for cutting are operable to cut the video footage automatically using a cutting algorithm.

5. A system as claimed in any one of the preceding claims, wherein the cutting means are operable to detect changes between frames of the video and define a cut when the changes exceed a pre-determined level.

6. A system as claimed in any one of the preceding claims, wherein the cutting means are operable to cut the video

content into clips on the basis of an intensity edge determination.

7. A system as claimed in any one of the preceding claims, wherein the cutting means comprises a histogram algorithm.

8. A system as claimed in any one of the preceding claims, wherein the cutting means are operable to cut the video content into clips in response to a user command.

9. A system as claimed in any one of the preceding claims comprising means for manipulating clips in response to a user command.

10. A system as claimed in claim 8, wherein the means for manipulating are operable to merge selected clips in response to a user command.

11. A system as claimed in claim 8, wherein the means for manipulating are operable to divide a selected clip in response to a user command.

12. A system as claimed in any one of the preceding claims comprising means for classifying the clips according to pre-determined classification criteria.

13. A system as claimed in claim 11, wherein the classifying means are operable to classify the clips according to any one or more of camera motion; optical flow; image

quality; scene changes; photometric characteristics; level of blurring and contrast detection.

14. A system as claimed in claim 11 or claim 12, wherein the classifying means are operable to classify a clip in response to a user defined classification.

15. A system as claimed in any one of claims 12 to 14, wherein the sorting means are operable to sort the clips into groups based on clip classifications.

16. A system as claimed in any one of the preceding claims comprising means for attributing a label descriptor to each clip.

17. A system as claimed in claim 16, wherein the attributing means are operable to automatically attribute a label descriptor to a clip.

18. A system as claimed in claim 16, wherein the attributing means are operable to attribute a label descriptor to a clip in response to a user selection.

19. A system as claimed in any one of the preceding claims, wherein the means for sorting are operable to sort the clips into one or a set of folders.

20. A system as claimed in any one of the preceding claims, wherein the means for sorting are operable to include each clip in a plurality of different groups.

21. A system as claimed in any one of the preceding claims, wherein the means for sorting are operable to sort the clips in response to a user selection.

22. A computer program, preferably on a data carrier or a computer readable medium, the computer program being for editing digital video footage using a computer processor, the computer program comprising code or instructions for cutting the digital video footage into a plurality of clips and sorting the clips into a plurality of groups.

23. A computer program as claimed in claim 22, comprising code or instructions for presenting on a screen or display a group identifier for each group of clips; receiving a user selection of a desired group identifier and presenting clip identifiers for clips included in the selected group in response to the user selection.

24. A computer program as claimed in claim 22 or claim 23, comprising code or instructions for searching for groups of clips based on user input criteria.

25. A computer program as claimed in any one of claims 22 to 23, wherein the code or instructions for cutting are operable to cut the video footage automatically using a cutting algorithm.

26. A computer program as claimed in any one of claims 22 to 25, wherein the code or instructions for cutting are operable to detect changes between frames of the video and define a cut when the changes exceed a pre-determined level.

27. A computer program as claimed in any one of claims 22 to 26, wherein the code or instructions for cutting are operable to cut the video content into clips on the basis of an intensity edge determination.

28. A computer program as claimed in any one of claims 22 to 27, wherein the code or instructions for cutting comprises a histogram algorithm.

29. A computer program as claimed in any one of claims 22 to 28, wherein the code or instructions for cutting are operable to cut the video content into clips in response to a user command.

30. A computer program as claimed in any one of claims 22 to 29 comprising code or instructions for manipulating or modifying clips in response to a user command.

31. A computer program as claimed in claim 30, wherein the code or instructions for manipulating are operable to merge selected clips in response to a user command.

32. A computer program as claimed in claim 30, wherein the code or instructions for manipulating are operable to divide a selected clip in response to a user command.

33. A computer program as claimed in any one of claims 22 to 33 comprising code or instructions for classifying the clips according to pre-determined classification criteria.

34. A computer program as claimed in claim 33, wherein the code or instructions for classifying are operable to classify the clips according to any one or more of camera motion detection; optical flow; image quality; scene changes; photometric characteristics; level of blurring and contrast detection.

35. A computer program as claimed in claim 33 or claim 34, wherein the code or instructions for classifying are operable to classify a clip in response to a user defined classification.

36. A computer program as claimed in any one of claims 33 to 35, wherein the code or instructions for sorting are operable to sort the clips into groups based on clip classifications.

37. A computer program as claimed in any one of claims 22 to 36 comprising code or instructions for attributing a label descriptor to each clip.

38. A computer program as claimed in claim 37, wherein the code or instructions for attributing are operable to automatically attribute a label descriptor to a clip.

39. A computer program as claimed in claim 37, wherein the code or instructions for attributing are operable to attribute a label descriptor to a clip in response to a user selection.

40. A computer program as claimed in any one of claims 22 to 39, wherein the code or instructions for sorting are operable to sort the clips into one or a set of folders.

41. A computer program as claimed in any one of claims 22 to 40, wherein the code or instructions for sorting are operable to include each clip in a plurality of different groups.

42. A digital camera or the like that includes a system or computer program as claimed in any one of the preceding claims, the digital camera being operable to provide the video footage for editing.

43. A method for editing digital video footage using a computer processor, the method comprising cutting the digital video footage into a plurality of clips and sorting the clips into a plurality of groups.

44. A method as claimed in claim 43, comprising presenting on a screen or display a group identifier for each group

of clips; receiving a user selection of a desired group identifier and presenting clip identifiers for clips included in the selected group in response to the user selection.

45. A method as claimed in claim 43 or claim 44, comprising searching for groups of clips based on user input criteria.

46. A method as claimed in any one of claims 43 to 45, wherein the step of cutting is done automatically using a cutting algorithm.

47. A method as claimed in any one of claims 43 to 46, wherein the step of cutting involves detecting changes between frames of the video and defining a cut when the changes exceed a pre-determined level.

48. A method as claimed in any one of claims 43 to 47, wherein the step of cutting involves cutting the video content into clips on the basis of an intensity edge determination.

49. A method as claimed in any one of claims 43 to 48, wherein the step of cutting comprises using a histogram algorithm.

50. A method as claimed in any one of claims 43 to 49, wherein the step of cutting involves receiving a user

command and cutting the video content into clips in response to the user command.

51. A method as claimed in any one of claims 43 to 50, comprising manipulating or modifying clips in response to a user command.

52. A method as claimed in claim 51, wherein the step of manipulating comprises merging selected clips in response to a user command.

53. A method as claimed in claim 51, wherein the step of manipulating involves dividing a selected clip in response to a user command.

54. A method as claimed in any one of claims 43 to 53 comprising classifying the clips according to pre-determined classification criteria.

55. A method as claimed in claim 54, wherein the step of classifying comprises classifying the clips according to any one or more of camera motion detection; optical flow; image quality; scene changes; photometric characteristics; level of blurring and contrast detection.

56. A method as claimed in claim 54 or claim 55, wherein the step of classifying comprises classifying a clip in response to a user defined classification.

57. A method as claimed in any one of claims 54 to 56, wherein the step of sorting is done based on clip classifications.

58. A method as claimed in any one of claims 43 to 57 comprising attributing a label descriptor to each clip.

59. A method as claimed in claim 58, wherein the step of attributing comprises automatically attribute a label descriptor to a clip.

60. A method as claimed in claim 58, wherein the step of attributing comprises attributing a label descriptor to a clip in response to a user selection.

61. A method as claimed in any one of claims 43 to 60, wherein the step of sorting comprises sorting the clips into one or a set of folders.

62. A method as claimed in any one of claims 43 to 61, wherein the step of sorting comprises including each clip in a plurality of different groups.

63. A method as claimed in any one of claims 43 to 62 comprising adding clips to the groups.

64. A method as claimed in claim 63 wherein the step of adding is done automatically.

65. A method as claimed in claim 63 wherein the step of adding is done in response to a user command.

66. A method as claimed in any one of claims 43 to 65 comprising creating new groups.

67. A method as claimed in claim 66 wherein the step of creating is done automatically.

68. A method as claimed in claim 67 wherein the step of creating is done in response to a user command.

69. A method as claimed in any one of claims 43 to 68 comprising using selected video clips to provide video content.

70. A method as claimed in claim 69 comprising copying the video content.

71. Video content that is produced using the system or computer program or method of any one of the preceding claims.

72. Copies of video content that is produced using the system or computer program or method of any one of the preceding claims.

73. A system substantially as described hereinbefore with reference to the accompanying drawings.

74. A computer program substantially as described hereinbefore with reference to the accompanying drawings.

75. A method substantially as described hereinbefore with reference to the accompanying drawings.

The Patent Office

PATENTS • DESIGNS • COPYRIGHT • TRADE MARKS

INVESTOR IN PEOPLE

52

Application No: GB 0208795.5
Claims searched: All

Examiner: Donal Grace
Date of search: 22 July 2003

Patents Act 1977 : Search Report under Section 17

Documents considered to be relevant:

Category	Relevant to claims	Identity of document and passage or figure of particular relevance	
X	1 to 72	EP 0780844 A2	(SIEMENS) see page 4 lines 26-27, page 12 line 46 to page 18 line 18
X E	1 to 72	GB 2380599 A	(MUVEE) see figures 16 and 17 and related description
X	1, 2, 3, 8, 9, 10, 11, 12, 14, 15, 16, 19, 21, 22, 23, 24, 33, 35, 37, 43, 44, 54, 56, 58, 71, 72 at least	EP 0915469 A2	(MATSUSHITA) see page 6 lines 36 to 46, page 7 lines 33 to 45, page 8 lines 1 to 26
X	1, 2, 3, 7, 12, 13, 22, 23, 28, 33, 34, 43, 44, 49, 54, 55 71, 72 at least	EP 0675495 A2	(SIEMENS) see page 5 lines 1 to 19, page 8 lines 24 to page 9 line 25
X	1, 2, 3, 4, 5, 6, 16, 22, 23, 25, 26, 27, 37, 43, 44, 46, 47, 48, 58, 71, 72 at least	US 5177513	(SAITO) see column 1 lines 34 to 42, column 6 lines 21 to 51, column 7 line 3 to column 8 line 64
X	1, 4, 5, 6, 22, 25, 26, 27, 43, 46, 47, 48, 71, 72 at least	US 5956453	(YAEGASHI) see abstract

Application No: GB 0208795.5
Claims searched: All

Examiner: Donal Grace
Date of search: 22 July 2003

Category	Relevant to claims	Identity of document and passage or figure of particular relevance
X	1, 4, 5, 6, 22, 25, 26, 27, 43, 46, 47, 48, 71, 72 at least	US 5339166 (LEBRAT) see column 6 lines 7 to 42, column 27 lines 44 to 60

Categories:

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

Field of Search:

Search of GB, EP, WO & US patent documents classified in the following areas of the UKC^{V}:

G5R

Worldwide search of patent documents classified in the following areas of the IPC^{7}:

G11B

The following online and other databases have been used in the preparation of this search report:

Online: EPODOC; JAPIO; WPI. Selected publications.

PUB-NO: GB002387710A
DOCUMENT-IDENTIFIER: GB 2387710 A
TITLE: Digital video editing
PUBN-DATE: October 22, 2003

INVENTOR-INFORMATION:

NAME	**COUNTRY**
ARREDONDO, MIGUEL	GB
MARSLAND, DOUGLAS	GB
LANE, DAVID MICHAEL	GB
OLMOS, ADRIANA	CA
CORMACK, ALASTAIR	GB
SCHMIDT, ARNAUD	FR
LEBART, KATIA	GB

ASSIGNEE-INFORMATION:

NAME	**COUNTRY**
UNIV HERIOT WATT	GB

APPL-NO: GB00208795
APPL-DATE: April 17, 2002

PRIORITY-DATA: GB00208795A (April 17, 2002)

INT-CL (IPC): G11B027/028

EUR-CL (EPC): G11B027/034 , G11B027/28 , G11B027/34

US-CL-CURRENT: 707/E17.028

ABSTRACT:

CHG DATE=20040807 STATUS=C>A system for editing digital video footage using a computer processor. The system is configured to cut (102) the digital video footage (100) into a plurality of clips and sort (104) those clips into groups. A group identifier associated with each of the groups is presented on a screen or display (107). The groups can be selected by a user so that the clips contained therein can be used. The cutting can be done automatically using a cutting algorithm, eg based on a histogram, or by intensity edge determination. Clips may be classified according to predetermined criteria eg camera motion, optical flow, image quality, scene changes, photometric characteristics level of blurring and contrast detection.